IN THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the present application:

1. (Currently amended) A method comprising:

maintaining a plurality of processes of a particular service type in a processing system;

assigning a status to each of the processes;

assigning a unique process identifier to each of the plurality of processes;

causing each of the processes to monitor the other processes of said service type;

~~causing each of the processes to respond autonomously to a predetermined condition by changing its own status between active and non-active; and~~

causing said plurality of processes to interact with each other to establish a priority of status, such that each of said plurality of processes can alter the priority of another of said plurality of processes without the use of a master to enable said interaction or alteration of priority, wherein said priority is based on a value of [[an]] the unique process identifier assigned to each of said plurality of processes.

2. (Currently amended) A method as recited in claim 1, further comprising:

causing each of the processes to respond autonomously to a predetermined condition by changing its own status between active and non-active, wherein the predetermined condition involves another process of the particular service type.

3. (Original) A method as recited in claim 2, further comprising causing each of the processes independently to maintain a list of other participant processes in the processing system.

4. (Original) A method as recited in claim 3, wherein the plurality of processes includes an active process and a non-active process corresponding to the active process, each independently maintaining said list.

5. (Original) A method as recited in claim 4, wherein the non-active process can autonomously change its status to active in response to an event affecting the active process.

6. (Original) A method as recited in claim 1, further comprising:

causing each of the processes to send heartbeat messages to each other process; and

causing each of the processes to listen for heartbeat messages from other processes;

causing each of the processes to update its list of participant processes based on receipt of heartbeat messages from other processes; and

causing each of the processes to update its list of participant processes based on the lack of receipt of heartbeat messages from other processes from which heartbeat messages have previously been received.

7. (Currently amended) A method as recited in claim 1, ~~further comprising assigning a unique process identifier to each of the processes,~~ wherein each process determines its status based on its unique process identifier.

8. (Original) A method as recited in claim 7, wherein each process determines its status based on the value of its unique process identifier relative to the value of the unique identifier of each other process.

9. (Currently amended) A method comprising:

introducing a plurality of processes into a processing system, each of the processes having a service type;

assigning a status to each of the processes, each said status selected from among a plurality of prioritized statuses, including a primary status and a standby status, such that at least one of the processes is a primary process and at least one of the processes is a standby process for the primary process;

assigning a unique process identifier to each of the processes;

causing said plurality of processes to interact with each other to establish a priority of status, such that each of said plurality of processes can alter the priority of another of said plurality of processes without the use of a master to enable said interaction or alteration of priority, wherein said priority is based on a value of [[an]] the unique process identifier assigned to each of said plurality of processes; and

maintaining each of the processes so that each of the processes monitors its own status and the status of each other process of the same service type and can

change its status from standby to primary without the use of a master, in response to an external event relating to a process of said same service type.

10. (Original) A method as recited in claim 9, further comprising causing each of the processes to maintain a list of other participant processes in the processing system.

11. (Currently amended) A method as recited in claim 9, ~~further comprising assigning a unique process identifier to each of the processes,~~ wherein each process determines its status based on its unique process identifier.

12. (Original) A method as recited in claim 11, wherein each process determines its status based on the value of its unique process identifier relative to the value of the unique identifier of each other process of the same service type.

13. (Currently amended) A method comprising:

introducing a plurality of processes into a processing system, each process having a service type;

causing each of the processes independently to maintain a list of other participant processes in the processing system.

assigning a unique process identifier to each of the processes;

causing each of the processes to send a heartbeat message repeatedly to each other process;

causing each of the processes to listen for heartbeat messages from other processes;

causing each of the processes to update its list of participant processes based on receipt of heartbeat messages from other processes;

causing each of the processes to update its list of participant processes based on the lack of receipt of heartbeat messages from other processes from which heartbeat messages have previously been received;

enabling each of the processes to select a status for itself, from among a plurality of prioritized statuses, including a primary and a standby status, without the use of a master, such that the plurality of processes includes a primary process and a standby process for the primary process; and

causing said plurality of processes to interact with each other to establish a priority of status, such that each of said plurality of processes can alter the priority of another of said plurality of processes without the use of a master to enable said interaction or alteration of priority, wherein said priority is based on a value of [[an]] the unique process identifier assigned to each of said plurality of processes.

14. (Original) A method as recited in claim 13, wherein for each process, the selection of status is based on the value of the unique process identifier of said process relative to the value of the unique process identifier of other processes having the same service type as said process.

15. (Currently amended) A processing system comprising:

a plurality of processes, each process having a service type;

means for assigning a status to each of the processes, each said status selected from among a plurality of prioritized statuses, including an active status and a standby status, such that at least one of the processes is a primary process and at least one of the processes is a standby process for the primary process;

means for assigning a unique process identifier to each of the processes;

means for maintaining each of the processes so that each of the processes monitors its own status and the status of each other process of the same service type and can autonomously change its status from standby to primary in response to an external event; and

means for causing said plurality of processes to interact with each other to establish a priority of status, such that each of said plurality of processes can alter the priority of another of said plurality of processes without the use of a master to enable said interaction or alteration of priority, wherein said priority is based on a value of [[an]] the unique process identifier assigned to each of said plurality of processes.

16. (Original) A processing system as recited in claim 15, further comprising means for causing each of the processes to maintain a list of other participant processes in the processing system.

17. (Currently amended) A processing system as recited in claim 15, ~~further comprising means for assigning a unique process identifier to each of the processes,~~ wherein each process determines its status based on its unique process identifier.

18. (Original) A processing system as recited in claim 15, wherein each process determines its status based on the value of its unique process identifier relative to the value of the unique identifier of each other process of the same service type.

19. (Currently amended) A method comprising:

maintaining a plurality of processes in a processing system, each process having an ability to independently monitor a status of each other process of said plurality of processes, without the use of a master;

assigning a unique process identifier to each of the plurality of processes; and

causing said plurality of processes to interact with each other to establish a priority of status, such that each of said plurality of processes can alter the priority of another of said plurality of processes without the use of a master to enable said interaction or alteration of priority, wherein said priority is based on a value of [[an]] the unique process identifier assigned to each of said plurality of processes.

20. (Original) A method as recited in claim 19, wherein said interaction and said alteration amongst said plurality of processes is used to enable fault tolerance for at least one of said processes in said processing system.

21. (Original) A method as recited in claim 19, wherein said status is one of: primary, to become primary, or standby.

22. (Canceled).

23. (Previously presented) A method as recited in claim 19, wherein said priority is further based on the status assigned to each of said plurality of processes.

24. (Currently amended) A method for providing fault tolerance in a processing system, the method comprising:

enabling a plurality of processes in a processing system each to broadcast a periodic heart-beat message, wherein said heart-beat message includes an identifier for each of said plurality of processes;

enabling each of said plurality of processes to receive each said heart-beat message;

causing each of said plurality of processes to maintain an individual record of said plurality of processes;

causing each of said plurality of processes to update said individual record based on said heart-beat messages;

assigning each of said processes with a status, wherein said status is one of: primary, to become primary, or standby;

assigning a unique process identifier to each of said plurality of processes;

enabling said plurality of processes to negotiate a hierarchy of control amongst each other based on the broadcast and receipt of heart-beat messages by each of said plurality of processes, wherein said hierarchy of control is based on the status of each of said plurality of processes; and

causing said plurality of processes to interact with each other to establish a priority of status, such that each of said plurality of processes can alter the priority of

another of said plurality of processes without the use of a master to enable said interaction or alteration of priority, wherein said priority is based on a value of [[an]] <u>said unique process</u> identifier assigned to each of said plurality of processes.